A PRE-SYNOPSIS

HYBRID REINFORCEMENT LEARNING CHAIN SYSTEM IN DISTRIBUTED BLOCKCHAIN

Submitted to the

MGM University

Chhatrapati Sambhajinagar (MS) - 431003, INDIA

For the Award of the Degree

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE & ENGINEERING

By

MONALI NITESH SHETTY

(PH21CSE23)

Under Guidance of

Dr. Sharvari Tamane

Professor & Head
Dept. of Information Technology, Jawaharlal Nehru Engineering College (JNEC) &
University Department of Information and Communication Technology (UDICT)
MGM University

December 2024

DECLARATION

I hereby declare that, the present work completed in the form of thesis entitled, "Hybrid Reinforcement Learning Chain System in Distributed Blockchain" is an original work carried out by me under the guidance of Dr. Sharvari Tamane, and has not been submitted or published in any form for the fulfilment of any other degree or any other similar to this in other University.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis.

Monali N. Shetty

RESEARCH SCHOLAR NAME AND SIGN

PLACE: CHATRAPATI SAMBHAJINAGAR

DATE:

MGM University

Chhatrapati Sambhajinagar

CERTIFICATE

This is to certify that Monali N. Shetty has carried out the work embodied in this thesis entitled "**Hybrid Reinforcement Learning Chain System in Distributed Blockchain**". The work included in this thesis is original, unless stated otherwise and has not been submitted for other degree of MGM University or any other University. References made to the work of others have been cited in the text.

SIGNATURE AND NAME OF RESEARCH SUPERVISOR

INDEX

Particulars	Page No
Introduction	1
Summary of Literature Survey and Research Gap	2
Research Objectives and Scope	3
Contributions	4
Proposed Methodology (PART 1 and PART 2)	5,11
Results (PART 1 and PART 2)	9, 14,15
Conclusion	16
Organisation of the Thesis	17
References	19
List of Publications and Conference Attended	21

INTRODUCTION

Blockchain technology is revolutionizing digital transactions by offering unparalleled security, transparency, and efficiency. For example, in 2021, the global blockchain market was valued at $4.93 billion and is projected to grow to $228 billion by 2028, highlighting its transformative impact on industries worldwide. However, as blockchain becomes the foundation for Web3 and the Metaverse, its security against cyber threats is paramount. This research aims to develop a **Hybrid Reinforcement Chain Learning System** to detect and mitigate cyber threats within a distributed blockchain environment focusing on ransomware, money laundering, and smart contract vulnerabilities.

Blockchain technology was first introduced by Satoshi Nakamoto as a distributed decentralized storage ledger for Bitcoin to overcome the problem of double spending [1]. As a decentralized digital ledger, blockchain records transactions across multiple computers, making it virtually impossible to alter or tamper with data. A blockchain is essentially a growing list of records, called blocks, which are linked together using cryptography. Each block contains a timestamp and a link to the previous block. Once a block of data is added to a blockchain, it is extremely difficult to change it [2]. Its unique attributes, such as immutability and distributed consensus, have sparked interest in applications far beyond digital currencies, extending into fields like supply chain management, finance, IoT, and cloud services.

The rapid growth of cryptocurrencies has posed significant challenges in the fight against cybercrime, especially the growing prevalence of money laundering and ransomware connections in cryptocurrencies such as Bitcoin in particular [3, 4]. Ransomware is a form of malicious software that encrypts a user's data, requiring a ransom—usually in Bitcoin—to unlock it. Cryptocurrencies like Bitcoin offer anonymity and decentralization, which makes it difficult to apply traditional financial regulations such as know your customers (KYC) and know your business (KYB) decisions. These characteristics has made cryptocurrencies attractive to cybercriminals due to money laundering and illegal storage of funds [5]. As a result, there is an urgent need for sophisticated methods to detect and prevent illegal activities [6].

In addition to these challenges, Ethereum's introduction of **smart contracts**—self-executing programs on blockchain networks—has expanded blockchain's use cases but introduced new security risks. These contracts, while revolutionary, are often targeted by attackers due to their immutability and the significant financial value they handle. Smart contract code is publicly visible, and the anonymity provided by blockchain makes it difficult to track down malicious actors once an attack has been executed. While there has been some prior research on smart contract security ([7,8]) and substantial work on their utility [9], this remains a rapidly evolving field. As blockchain technology and smart contracts continue to develop, the need for more comprehensive understanding and coverage of security issues grows. To stay current and effectively tackle smart contract security issues, it's important to combine the latest open-source information with well-established academic research.

SUMMARY OF LITERATURE SURVEY AND RESEARCH GAP

A comprehensive literature review was conducted to identify gaps in existing blockchain security solutions. To summarize the trends observed from literature, many studies used supervised learning methods, but challenges like dataset shortage or adaptability to new attack vectors remain common.

Research Gap

1. Limited anomaly detection for novel threats – Adapt to changes in the transaction patterns
2. There is a lack of thorough research focused on adaptive security models that address how the gas mechanisms in Ethereum smart contracts operate and their implications for security.
3. Vulnerability detection models depend on smart contract source code or bytecode and not on metadata.
4. Balancing exploration and exploitation in reinforcement learning

RESEARCH OBJECTIVES AND SCOPE

Objectives of Research

The primary objective of this research is to develop a robust hybrid reinforcement learning chain system for detecting anomalies in distributed blockchain networks. It aims to:

1. To derive attention towards the security vulnerabilities of Blockchain systems via a systematic and comprehensive study
2. To solve the problem of shortage and imbalanced classes in a Dataset
3. To develop a Deep Reinforcement Learning approach for detecting ransomware transactions in Bitcoin, with a focus on improving accuracy using Boltzmann Exploration.
4. To detect DoS attacks related to gas consumption in the Ethereum network(metadata) using Deep Reinforcement Learning with Epsilon Greedy Exploration.
5. To detect Gas Griefing attacks in the Ethereum network (metadata) using Deep Reinforcement Learning with Epsilon Greedy Exploration.
6. To evaluate reinforcement learning models for adapting to evolving cyberattack patterns, balancing exploration and exploitation in blockchain anomaly detection.

Scope

1. Investigating current machine learning and neural network techniques for detecting ransomware, DoS, and gas griefing attacks.
2. Designing and implementing a hybrid reinforcement learning system that can dynamically adjust to changing attack patterns.
3. Analyzing the performance of the system compared to existing solutions in terms of detection accuracy, adaptability, and scalability.

CONTRIBUTIONS

***Contributions*:** This research's significant contributions are listed below:

(i) RL-based Deep Q-Network (DQN) detectors are proposed to detect ransomware payment transactions in Blockchain Bitcoin network.

(ii) The ability of the Deep Reinforcement Learning detector to adapt to changes in the payment transaction patterns is investigated.

(iii) Rigorously tests and evaluates Deep Reinforcement Learning with Epsilon Greedy Exploration and Boltzmann Exploration to learn the distinctive transaction patterns in Bitcoin payment transactions.

(iv) We have provided comprehensive understanding of the features of the Ethereum blockchain, including the gas allocation processes and vulnerabilities revealed by EIP-150.

(v) RL-based Deep Q-Network (DQN) with Epsilon Greedy Exploration detectors are proposed to detect DOS attack in smart contract of Blockchain Ethereum network.

(vi) Reinforcement Learning based DQN detectors are proposed to detect Gas Griefing attack in smart contract and compared it to traditional machine learning methods.

(vii) Thoroughly tests and evaluates deep reinforcement learning with Epsilon Greedy exploration to learn the malicious smart contracts in Ethereum.

PROPOSED METHODOLOGY

PART 1 - Detection of Ransomware Transaction in Bitcoin

1.Problem Formulation

Importance of Detecting Ransomware Transactions

Ransomware attacks represent a critical threat to the security of Bitcoin transactions. Identifying these transactions early allows individuals, businesses, and institutions to mitigate financial losses, data breaches, and operational disruptions. As Bitcoin's decentralized and pseudonymous nature facilitates anonymous ransom payments, tracking and monitoring ransomware transactions is essential to dismantle criminal networks and protect victims.

In this section, we will provide a more detailed and critical analysis of the proposed detection methodology and the challenges that remain unaddressed. This improved approach will incorporate reinforcement learning with dynamic exploration strategies to optimize ransomware detection in Bitcoin.

2. Pre-processing of Dataset

The Bitcoin transaction dataset used in this study contains a significant class imbalance, with 2,875,284 non-malicious transactions compared to only 41,413 ransomware-related addresses [24]. This imbalance hinders accurate anomaly detection and ransomware identification. To address this, the study implements a combination of **SMOTE** (Synthetic Minority Over-Sampling Technique) and **RUS** (Random Under-Sampling). SMOTE generates synthetic samples for the minority class, improving the model's ability to detect ransomware transactions. However, oversampling can cause the majority class to dominate, so RUS is applied to reduce non-malicious transactions, preventing overfitting. Combining SMOTE and RUS optimizes the dataset's balance, enhancing the machine learning model's performance.

3. Proposed Reinforcement Learning Approach

Figure.1. displays the working for our proposed methodology.

The primary goal of this work is to create an automated system that can recognize ransomware-style cyberattacks on end devices using Q-learning reinforcement learning. Using the Q-tab or state-action values for self-directed learning, a reinforcement learner (Q-learner) is employed in the proposed architecture, as illustrated in Figure.1, to teach the environment to do actions in the future.

In the proposed novel Boltzmann exploration technique, the agent makes probabilistic action choices depending on temperature parameter and estimated action values (Q-values). The degree of exploration versus exploitation is controlled by the temperature parameter. More exploratory behaviour

is produced by a high temperature value, whereas more exploitative behaviour is produced by a low temperature value [11].

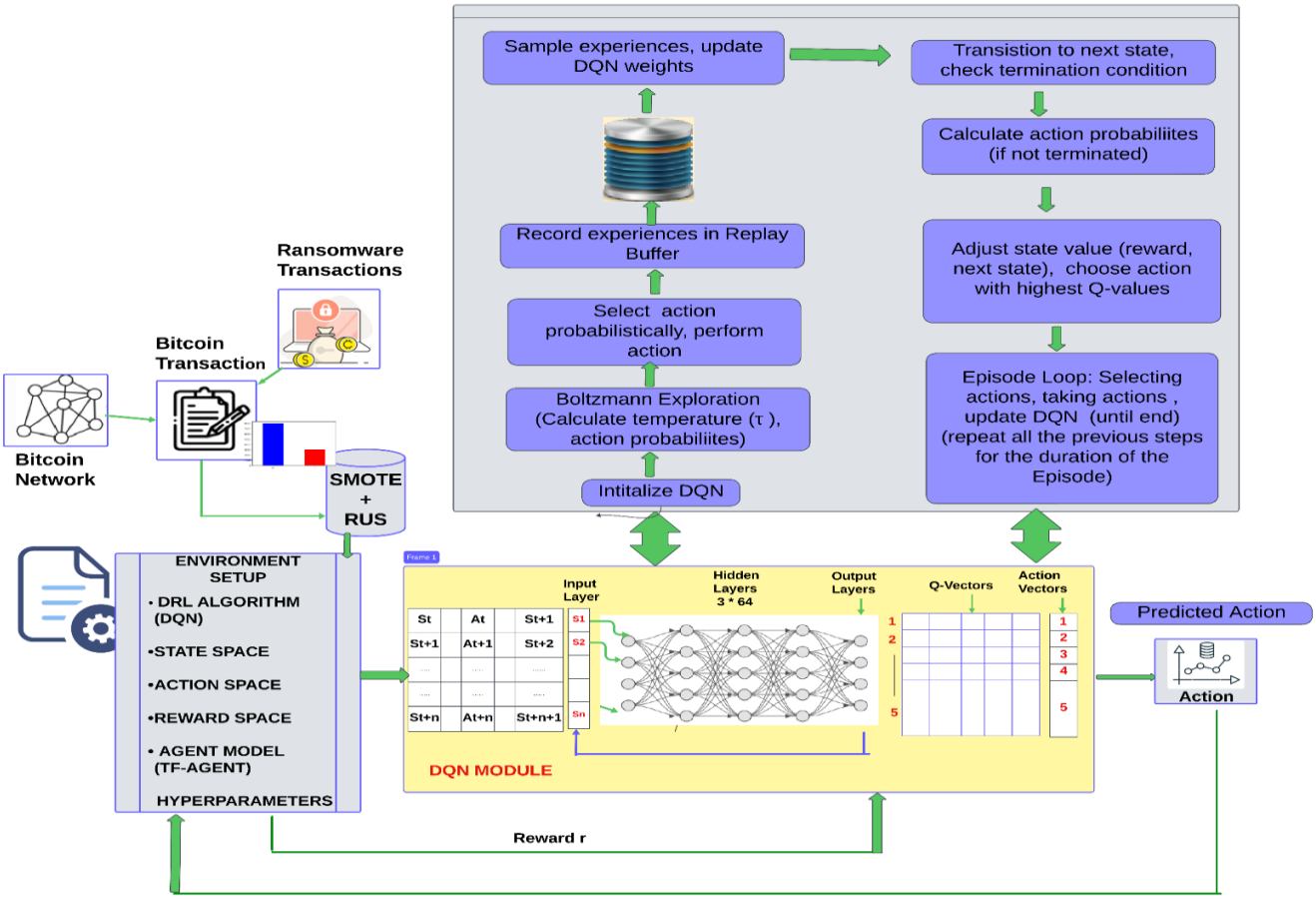

Sample experiences, update DQN weights
Transistion to next state, check termination condition
Calculate action probabiliites (if not terminated)
Record experiences in Replay Buffer
Adjust state value (reward, next state), choose action with highest Q-values
Select action probabilistically, perform action
Episode Loop: Selecting actions, taking actions , update DQN (until end) (repeat all the previous steps for the duration of the Episode)
Boltzmann Exploration (Calculate temperature (τ), action probabiliites)
Intitalize DQN
Ransomware Transactions
Bitcoin Transaction
Bitcoin Network
SMOTE + RUS
ENVIRONMENT SETUP
• DRL ALGORITHM (DQN)
•STATE SPACE
•ACTION SPACE
•REWARD SPACE
• AGENT MODEL (TF-AGENT)
HYPERPARAMETERS
Input Layer
Hidden Layers 3 * 64
Output Layers
Q-Vectors
Action Vectors
DQN MODULE
Predicted Action
Action
Reward r

Figure. 1. Architecture Diagram of Proposed Methodology

1. Estimating Action Values: Reinforcement learning aims to teach an agent a policy that maximizes its predicted cumulative reward over a predefined period of time. The agent keeps estimate of the values of various actions in different states in order to accomplish this. These values show how beneficial it is to do a specific activity in a specific situation.
2. Boltzmann Distribution: By using the Boltzmann distribution, Boltzmann exploration assigns probabilities to actions based on estimated values. The following formula is used to determine the probability of choosing a particular action in a given state:

$$P(a) = exp(\frac{Q(a)}{\tau}) / \sum = \ [\![(exp(Q(a\_i)/\tau)]\!] \qquad (1)$$

where

- The probability of choosing action 'a' is denoted by 'P(a)'.
- The estimated value of doing action 'a' in the current situation is denoted by 'Q(a)'.

- ‘τ’ is the parameter known as the ‘temperature’. It controls the level of exploration, where higher value of ‘τ' make the distribution more uniform (encouraging exploration) and lower values make it more focused (encouraging exploitation).
- The denominator Σ represents the sum of exponentiated values of estimated action qualities for all possible actions **a_i**.

3. Balancing Exploration and Exploitation: In Boltzmann exploration, one crucial parameter is the temperature, or τ. it enables the RL agent to strike a balance between exploring new actions and utilizing existing knowledge. Higher values of τ encourage the agent to explore new options, even if they do not appear to be optimal based on existing knowledge. The agent emphasizes exploitation by favouring actions with larger estimated values when τ is lower.
4. Adaptive Exploration: The versatility of Boltzmann exploration is one of its benefits. Setting a high τ allows the agent to explore broadly at first, when it knows very little about the environment around it. As the agent gains experience and gains more knowledge about the environment, τ can be gradually decreased, which will increase the exploitation of the most well-known activities. This flexibility is useful in situations where conditions are changing quickly.
5. Exploration Strategies: In RL, Boltzmann exploration is merely one type of exploration tactic. Other approaches, including ε-greedy exploration (in which the agent chooses the most well-known action with probability 1-ε and explores with probability ε), are equally useful and are selected according to the objectives and particular challenge.

This Boltzmann exploration method works especially well in situations where the agent must learn new behaviours while using the information it already has to make wise choices.

Action space and adversarial model

In order to determine states, we took into account the dataset's features length, looped, income, and label. Calculate the values of the state variables Label-Lb, income-I, looped-L1, and length-L for five rows. There are two approaches to dealing with a ransomware attack: alert (A) or ignore (I). Reward guidelines for

- length, if longer length (>100) = +1 (A) else -1 (I),
- looped, if high looped count (>1000) = +1 (A) else -1(I),
- income, if high or rapidly increasing income (>600000000000) = +1 (A) else -1(I)

Here we have considered two actions, A - Alert and I-Ignore and rewards are +1 for correct action and -1 for wrong action.

For ex. length as a state,

a1. Correct Classification (Non-Malicious):

- Reward: +1.0
- Condition: The agent correctly classifies a non-malicious (benign) transaction with a longer length as "non-Malicious."

a2. ***Correct Classification (Malicious)****:*

- Reward: +1.0
- Condition: The agent correctly classifies a malicious (ransomware) transaction with a longer length as "Malicious."

a3. ***Incorrect Classification (Non-Malicious)****:*

- Reward: -1.0
- Condition: The agent incorrectly classifies a non-malicious transaction with a longer length as "Malicious."

a4. Incorrect Classification (Malicious):

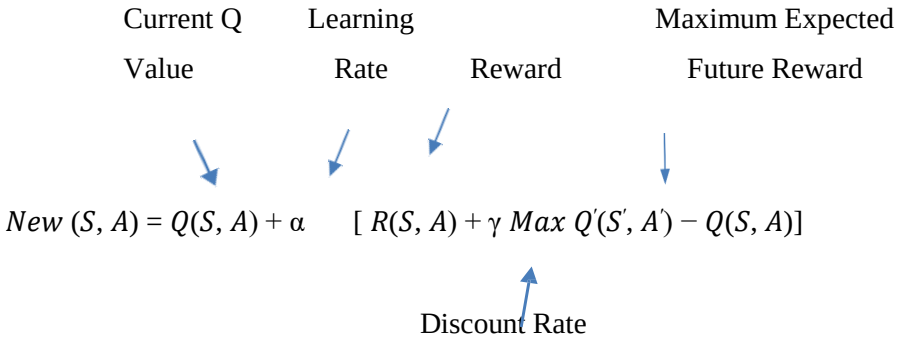
- Reward: -1.0
- Condition: The agent incorrectly classifies a malicious transaction with a longer length as "non-Malicious."

a5. Neutral or Uninformative Actions:

- Reward: 0.0
- Condition: The agent selects an action that doesn't lead to any classification (e.g., taking no action or choosing not to classify).

Record values in the R table in accordance with the action-reward policy. Use the Q formula. Fill in the Q-table with values. We may always look up the Q-table to see if an attack is likely to occur. Because there was no attack at first, values in the Q table were negative. But after a few transactions, values began to increase, indicating that the model had become aware of the attack. There won't be an attack if I (ignore) values are higher positives. Attack potential exists if A (Alert) levels are smaller negative. We are able to provide an attack result or no attack after all entries have been calculated.

We present a Deep Reinforcement Learning (DRL) system with 4 states. The Bellman equation,


Current Q Value
Learning Rate
Reward
Maximum Expected Future Reward
$New\ (S, A) = Q(S, A) + \alpha \quad [\ R(S, A) + \gamma\ Max\ Q'(S', A') - Q(S, A)]$
Discount Rate

is utilised as the value function in a reinforcement learning algorithm with Q-learning. The formula states that a state-action pair's Q value is equal to the sum of its possible future Q-values and current reward. In short, the exact reward is added to possible rewards. It's a discrete equation. However, most real-world applications involve continuous states and activities. Consequently, an effective function approximation technique is needed for the value function. This is when neural networks come into work. Hidden layers of neural networks are used to calculate each state and Q-value in the value function.

4. Experimental Setup

Establishing the environment for the detection of ransomware transactions in Bitcoin via the utilization of OpenAI Gym and TensorFlow Agents requires installing and configuring the tools and frameworks necessary for implementing and testing the deep reinforcement learning model.

Table 1, displays the parameters and values related to the Deep Reinforcement Learning model. Aside from the input and output layers, we employed three hidden layers with the "Relu" activation function for the fully connected architecture.

5. Results

Table 1: Performance evaluation of the proposed methodology (DRL with Boltzmann Exploration) based on various factors like learning rate, batch size and discount factor values.

Learning Rate	Gamma (Discount factor) γ	Batch size	Boltzmann Exploration (Temperature)	Training Accuracy	Validation Accuracy	F1 Score	Precision	Recall
	Learning rate varies							
0.001	0.99	32	0.5	91.33%	91.19%	0.93	0.89	0.97
0.01	0.99	32	0.5	50%	49%	0.65	0.49	1.0
	Batch Size varies							
0.001 0.001	0.99 0.99	32 64	0.5 0.5	91.33% 54%	91.19% 54%	0.93 0.63	0.89 0.511	0.97 0.84
0.001	0.99	128	0.5	47%	47%	0.63	0.477	0.94
	Gamma (Discount factor varies)							
0.001 0.001	0.10 0.99	32 32	0.5 0.5	72% 91.33%	72% 91.19%	0.75 0.93	0.63 0.89	0.93 0.97

Table 2: Boltzmann Exploration vs. Other Techniques in Bitcoin Transaction Analysis

Exploration Techniques	Exploration Rate	Learning Rate	Accuracy
Random Exploration (uniform random selection)	Random – action with equal probability from the entire space.	Next action random	63%
Epsilon- Greedy Exploration	Epsilon (Ɛ) =0.02	Alpha =0.001	83.70%
Proposed Boltzmann Exploration	Temperature (τ) =0.5	Alpha = 0.001	91%

Table 3: Performance Comparison of the Proposed Model under Diverse Attack Actions

DRL Exploration Technique	Testing Accuracy (diverse attack actions)	Standard Deviation
Random Exploration	61.35%	±1.2%
Epsilon Greedy Exploration	84.55%	±0.8%
Proposed Boltzmann Exploration	89.23%	±1.1%

Table 4: Overall Performance Comparison of the Proposed Model

DRL Exploration Technique	**Average Rewards over 100 Episodes**	**Success Rate**
Random Exploration	200	TP/P = 1229/1448 =85%
Epsilon Greedy Exploration	200	TP/P= 1874/2202 = 87.1 %
Boltzmann Exploration	200	TP/P = 3561/3709 = 95.7%

PART 2 - Detection of DOS Attacks in Ethereum

1. Problem Formulation

A major advancement in blockchain technology is the development of blockchain systems supporting smart contracts, e.g., Ethereum [12] and EOS [13]. Smart contracts are programs running on top of a blockchain that can receive and execute transactions autonomously. The growing interest in smart contracts is driven by their inherited decentralized nature that enables secure distributed computations while eliminating the need for trusted third parties [14]. Smart contracts, thus, provide means to automate and host decentralized applications (DApps) [15] and integrate other various innovations with the blockchain, with Ethereum currently being the most popular blockchain platform for running smart contracts. Ethereum smart contracts are written in high-level Turing-complete programming languages, such as Solidity, and they are compiled to Ethereum virtual machine (EVM) low-level bytecode that is deployed to the blockchain. The bytecode deployed on Ethereum permissionless public network is visible to the public and can be invoked by anyone who has an account on the blockchain [16]. Smart contract invocation requests from users are created as transactions that get added to the blockchain. Unlike traditional applications, smart contracts store state variables on the blockchain in a native persistent memory area(storage) maintained over executions. Further, the execution of smart contracts on the Ethereum blockchain consumes fees, called gas, paid for by users calling the contract functions. The gas consumed varies for each execution, and depends mainly on the executed bytecode instructions and the state of the contract on the blockchain. A contract execution proceeds as long as the users dedicate enough gas; within the limit set by Ethereum (i.e., block gas limit).

Smart contracts on Ethereum are susceptible to bugs and vulnerabilities, which can result in severe financial and reputational damage. For instance, the **DAO vulnerability** in 2016 led to the theft of $60 million worth of Ether [17], and an access control flaw in the **Value DeFi platform [18]** in 2021 allowed hackers to steal $10 million. More recently, **DeFi exploits** [19,20] in 2021 resulted in $1.55 billion in losses. These incidents highlight the importance of rigorous testing and monitoring of smart contracts to detect vulnerabilities before they are exploited. However, **immutability** in blockchain complicates efforts to reverse attack transactions or recover financial losses post-exploitation.

Therefore, it is critical to analyse and test smart contracts thoroughly for vulnerabilities before deployment. The immutability of blockchain transactions makes it difficult to reverse the attack transactions and recover the financial losses. Test cases can be helpful in detecting vulnerabilities in smart contracts, but they have limitations. The code coverage of test cases is limited due to the difficulty of covering all possible scenarios [21, 22]. Further, several corner cases get overlooked by testing, and it is difficult to reproduce all vulnerabilities in a test environment [23]. Additionally, the gas mechanism limits the adoption of runtime verification as instrumenting the code with assertions increases the gas cost.

To address these limitations, we propose a **dynamic monitoring framework** that relies solely on **transaction metadata and balance data** from the blockchain system. This approach eliminates the need for domain knowledge, source code access, or specialized execution environments, thereby significantly reducing gas costs associated with runtime verification. The key innovation lies in applying **Deep Reinforcement Learning (DRL)** to classify transactions as benign or harmful based on blockchain activity in real time.

2. Reinforcement Learning Approach

The proposed **Deep Q-Network (DQN) with Epsilon Greedy algorithm** for detecting DOS attacks works as:

Input Representation: The Deep Reinforcement Learning (DRL) agent receives a transaction as an input, which is represented as a feature vector. This vector encapsulates key characteristics of the transaction, such as:

Length: The size or number of operations in the transaction.
Looped: A binary feature indicating whether the transaction contains looping structures (a potential indicator of malicious activity).
Income: The financial gain associated with the transaction.
Label: An indicator of whether the transaction has been previously labelled as malicious or benign (for training purposes).

1. These features are fed into the input layer of the DQN, which processes them through a series of fully connected hidden layers to capture complex patterns in the data.
2. Using DQN :
   The DQN model learns a policy to classify the transaction as either malicious or benign based on the input features. During training, the agent performs the following steps:
   - Action Selection: The DQN agent is tasked with selecting an action at each decision point. In this context, the action space is binary, with two possible actions: classify the transaction as malicious or benign.
   - Policy: The policy $\pi(s, a)$ represents the agent's strategy for deciding which action (classification) to take in any given state (transaction feature set). This policy is learned by estimating the Q-value, which represents the expected cumulative reward for taking action a in state s.
   - Threshold: The output of the DQN is a probability distribution over the action space (malicious/benign). The agent computes a probability score indicating the likelihood of the transaction being malicious. If this probability exceeds a pre-defined threshold, the transaction is classified as malicious. Otherwise, it is classified as benign.
3. Reward Mechanism:
   The reward mechanism is critical to reinforcing the agent's learning behavior. The agent receives feedback in the form of rewards based on the accuracy of its classifications, which updates the Q-values and guides the learning process. The reward system works as follows:
   - Positive Reward: If the agent correctly classifies a malicious transaction, it is awarded a positive reward (e.g., +1). This incentivizes the agent to correctly identify malicious activities in future states.

- Negative Reward: If the agent incorrectly classifies a benign transaction as malicious or a malicious transaction as benign, it receives a negative reward (e.g., -1). This penalty discourages the agent from making false-positive or false-negative classifications.
- Exploration and Exploitation: To improve the agent's learning over time, an exploration strategy ε-greedy exploration is used. This ensures that the agent explores a wide range of possible actions (classifications) during training before converging on the optimal policy that maximizes the cumulative reward.

4. Training:
   - Replay Buffer: To stabilize learning, the DQN employs experience replay. The agent's past experiences (state, action, reward, next state) are stored in a replay buffer, from which random samples are drawn to train the network. This helps break the correlation between consecutive experiences, improving training stability.
   - Loss Function: The DQN minimizes the loss function, which is typically the mean squared error (MSE) between the predicted Q-values and the target Q-values. The target Q-values are updated using the Bellman equation, incorporating the observed reward and the maximum future reward. This allows the agent to improve its classification policy over time.

By continuously adjusting its policy based on feedback from the environment (rewards), the DQN agent learns to accurately classify transactions, maximizing the detection of DOS attacks while minimizing false classifications.

3. Results and Discussion

Table 5: Detection of DOS attack in Ethereum Smart Contract

Environment	**Models**	**Accuracy**
Non-Adversarial Environment	SVM Classifier	68.74%
Non-Adversarial Environment	Naïve Bayes	66.38%
Non-Adversarial Environment	Random Forest	64%
Non-Adversarial Environment	XGBoost	78.03%
Non-Adversarial Environment	LightGBM	84.91%
Non-Adversarial Environment	Voting Classifier	91.37%
Non-Adversarial Environment	AdaBoost	93.46%
Non-Adversarial Environment	CatBoost	95.07%
Non-Adversarial Environment	**Proposed Voting Classifier Model**	**91.37%**
Proposed Adversarial Environment	**Deep Reinforcement Learning (DRL- DQN) with Epsilon Greedy Exploration**	**95.68%**

Table 6: Performance Comparison of the Proposed Model and Existing Work

Algorithms	**Accuracy**	**Precision**	**Recall**	Dataset Used
GA based model [14]		77%	76%	Own dataset -BCCC-VolSCs-2023
CNN Model [26]	77%	77.50%	74.46%	Smart Contract Sanctuary [25]
Proposed Voting Classifier Model	91.37%	90%	90%	Own Dataset extracted from Etherscan.io
Deep Reinforcement Learning (DRL-DQN) with Epsilon Greedy Exploration	95.68%	88%	99%	Own Dataset extracted from Etherscan.io

Detection of Gas Griefing Attacks in Ethereum

1. Problem Formulation

The research examines the vulnerabilities that lead to gas grief attacks, specifically related to gas sharing mechanisms such as the 63/64 code described in Ethereum Improvement Proposal (EIP) 150. It provides detailed information on smart contract security challenges, introduces Gas Griefing attacks on Ethereum. This section explains the basics of Reinforcement Learning, and explores the implications of gas sharing rules.

Table 7. Comparison of Traditional Machine Learning Detectors in Non-Adversarial Environment vs. Proposed Detectors in Adversarial Environment

Environment	Algorithms	Accuracy	Precision	Recall	F1-score
Non-Adversarial Environment	SVM	81.15%	70%	76%	73%
Non-Adversarial Environment	Random Forest (RF)	89.05%	86%	79%	83%
Non-Adversarial Environment	LSTM	83.60%	76%	74%	75%
Adversarial Environment	**Deep Reinforcement Learning (DRL-DQN) with Epsilon Greedy Exploration**	**91.3%**	**86%**	**85%**	**87%**

Table 8: Performance Comparison of the Proposed Model in Adversarial Environment and Existing Work in Non-Adversarial Environment

Algorithms	Accuracy	Precision	Recall	F1-score
SVM	81.15%	70%	76%	73%
Random Forest (RF)	89.05%	86%	79%	83%
LSTM	83.60%	76%	74%	75%
Deep Reinforcement Learning (DRL- DQN) with Epsilon Greedy Exploration	**91.3%**	**86%**	**85%**	**87%**

CONCLUSION

This research successfully developed and evaluated innovative techniques for improving the security and integrity of blockchain networks, specifically focusing on ransomware detection in Bitcoin transactions and gas-related vulnerability detection in Ethereum smart contracts. By combining Deep Reinforcement Learning (DRL) with Boltzmann exploration, we were able to enhance the detection of ransomware in Bitcoin transactions, achieving high accuracy (91%) and effectiveness in identifying evolving ransomware patterns. This approach addresses key challenges such as limited training data, effective reward function design, and the balance between exploration and exploitation, demonstrating superior performance compared to traditional machine learning methods.

For Ethereum smart contracts, our research introduced a novel approach using Deep Q-reinforcement learning with Epsilon Greedy method to detect gas-related vulnerabilities. By modeling smart contracts as agents within a reinforcement learning framework, we provided a dynamic method for identifying potential security risks during contract execution. The application of Q-learning in this domain achieved a remarkable accuracy rate of 95.72%, surpassing existing static analysis methods and highlighting the potential for adaptive security measures in decentralized systems.

Overall, this study has shown that reinforcement learning techniques, especially when combined with advanced exploration strategies, can significantly improve blockchain security by offering adaptive, real-time detection mechanisms. These findings not only exceed the performance of current predictive algorithms but also open new pathways for future research into intelligent, autonomous systems for blockchain protection. As ransomware and gas-related attacks continue to evolve, our work lays the groundwork for more robust, scalable security solutions that can safeguard blockchain ecosystems, protect users, and combat cybercrime in decentralized environments.

ORGANISATION OF THE THESIS

TABLE OF CONTENTS

Chapter 1: Introduction
1. Overview
2. Motivation of Research
3. Thesis Organisation

Chapter 2: Concepts and Background
2.1 Blockchain Technology and Its Importance
- 2.1.1 Blockchain and Cryptocurrencies
- 2.1.2 Ethereum Platform
  - A. Ethereum Virtual Machine(EVM)
  - B. Smart Contracts

2.2 Security Challenges in Blockchain
2.3 The Need for Advanced Detection Mechanisms

Chapter 3: Literature Review
3.1 History of Blockchain Vulnerabilities and Attacks
3.2 Detection Methods of Attacks
3.3 Machine learning in Blockchain Security
3.4 Neural Networks in Blockchain Security
3.5 Reinforcement Learning Blockchain Security
3.6 Summary of Literature Review
3.7 Identification of Research Gaps

Chapter 4: Problem Statement, Objectives
4.1 Problem Statement
4.2 Objectives
4.3 Contributions

Chapter 5: Proposed Methodology
PART A: Detection of Ransomware Transaction Attacks in Bitcoin
5.1 Introduction
5.2 Problem Formulation
5.3 Machine Learning Approaches
5.4 Reinforcement Learning Approaches
5.5 Experimental Setup
5.6 Results and Discussion
5.7 Comparative Analysis
5.8 Summary

PART B: Detection of Gas Related Attacks (DOS Attack and Gas Griefing Attack) in Ethereum Smart Contract
6.1 Introduction
- 6.1.1 DOS Attack
- 6.2.2 Gas Griefing Attack

6.2 Problem Formulation
6.3 Machine Learning Approaches
6.4 Reinforcement Learning Approaches
6.5 Experimental Setup
6.6 Results and Discussion
6.7 Comparative Analysis
6.8 Summary

Chapter 7: Conclusion
7.1. Conclusion
7.2 Scope of Future Work

References

References

[1**]** S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Manubot, Tech. Rep., 2019

[2] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck. Blockchain. Bus. & Inf. Syst. Eng., vol. 59, no. 3, pp. 183–187, 2017.

[3] L.Y. Connolly, D. S. Wall, The Rise of Crypto-Ransomware in A Changing Cybercrime Landscape: Taxonomising Countermeasures. Computers& Security 87 (2019)101568.

[4] S. Kok, A. Abdullah, N. Jhanjhi, M. Supramaniam, Ransomware, Threat and Detection Techniques: A Review. Int. J. Comput. Sci. Netw. Secur 19(2) (2019)136.

[5] A. A. Badawi and Q. A. Al-Haija, "Detection of money laundering in bitcoin transactions," 4th Smart Cities Symposium (SCS 2021), Online Conference, Bahrain, 2021, pp. 458-464, doi: 10.1049/icp.2022.0387.

[6] S. A. Alsaif, Machine Learning-Based Ransomware Classification of Bitcoin Transactions. Applied Computational Intelligence and Soft Computing, 2023 (2023).

[7] P. Qian, Z. Liu, Q. He, R. Zimmermann, And X. Wang. “Towards Automated Re-entrancy Detection for Smart Contracts Based on Sequential Models”. In: IEEE Access 8 (2020), Pp. 19685–19695. DOI:10.1109/Access.2020.2969429.

[8] S. Sayeed, H. Marco-Gisbert, and T. Caira. “Smart Contract: Attacks and Protections”. In: IEEE Access 8 (2020), pp. 24416–24427. doi: 10.1109/ACCESS.2020. 2970495

[9] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock. “Blockchain technology in the energy sector: A systematic review of challenges and opportunities”. In: Renewable and Sustainable Energy Reviews 100 (2019), pp. 143–174. doi: 10.1016/j.rser.2018.10.014.

[11] Tong Wu, Jorge Ortiz. RLAD: Time Series Anomaly Detection through Reinforcement Learning and Active Learning. ACM ISBN 978-1-4503-9999-9/18/06, https://doi.org/10.1145/1122445.1122456, c 2021

[12] G. Wood et al. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Ethereum Project Yellow Paper, 151(2014):1–32, 2014. pages 1, 2, 4, 13

[13] E. IO. Eos. io Technical White Paper. EOS. IO (accessed 18 December 2017) https://github.com/EOSIO/Documentation, page 9, 2017.

[14] I. Grishchenko, M. Maffei, and C. Schneidewind. A Semantic Framework for The Security Analysis of Ethereum Smart Contracts. In International Conference on Principles of Security and Trust, pages 243–269. Springer, 2018.

[15] D. Johnston, S. O. Yilmaz, J. Kandah, N. Bentenitis, F. Hashemi, R. Gross, S. Wilkinson, and S. Mason. The General Theory of Decentralized Applications, Dapps. 2014. URL

[16] Ethereum. Networks. URL https://ethereum.org/en/developers/docs/networks

[17] P. Daian. Analysis of the DAO exploit, 2016. URL https://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit

[18] REKT. Value defi - rekt 2, 2021. URL https://rekt.news/value-rekt2

[19] Web. Decentralized Finance (DeFi), 2022. URL https://ethereum.org/en/defi

[20] S. M. Werner, D. Perez, L. Gudgeon, A. Klages-Mundt, D. Harz, and W. J. Knottenbelt. Sok: Decentralize Finance (Defi). Arxiv Preprint Arxiv:2101.08778, 2021

[21] B. Hailpern and P. Santhanam. Software Debugging, Testing, And Verification. IBM Systems Journal,41(1):4–12, 2002

[22] C. Kaner, J. Bach, and B. Pettichord. Lessons Learned in Software Testing. John Wiley & Sons, 2008

[23] J. Viega and G. R. Mcgraw. Building Secure Software: How to Avoid Security Problems the Right Way, Portable Documents. Pearson Education, 2001

[24] Bitcoin Heist Ransomware Address [Dataset]. (2020). UCI Machine Learning Repository. https://doi.org/10.24432/C5BG8V.

[25] Hu T, Liu X and Liu Y. Transaction-based classification and detection approach for Ethereum smart contract. *Information Processing & Management*. 2020; 58(4): 106-113. doi: 10.1016/j.ipm.2020.102462.

[26] Duan L, Yang L, Liu C, Ni W and Wang W. A New Smart Contract Anomaly Detection Method by Fusing Opcode and Source Code Features for Blockchain Services. *IEEE Transactions on Network and Service Management*. 2023; 20(4): 4354-4368. doi: 10.1109/TNSM.2023.3278311.

List of Publications and Conference Attended

1. **Journals:**
    - 2 Journals published (Scopus Indexed (Q3) and Web of Science Indexed)
    - 1 Journal in the publishing process (accepted) (SCIE Indexed) (Q1)
2. **Conference Papers:**
    - 1 Conference paper published (Springer)
    - 2 (Scopus Indexed Conference)
3. **Intellectual Property:**
    - 1 Canara Copyright granted
    - 1 UK Patent granted
    - 1 Indian Patent published
    - 2 Indian Copyright in the hearing stage

Journals:

1. "Unveiling Bitcoin Network Attack using Deep Reinforcement Learning with Boltzmann Exploration", Peer-To-Peer Networking and Applications, SPRINGER, SCIE, SCOPUS Indexed (Q2).

2. "Securing the Chain: A Systematic Review of Blockchain Systems with Emphasis on Anomaly Detection Strategies", TELEMATIQUE, Web of Science (WoS), ISSN: 1856-4194, Volume 22 Issue 1, 2023, 3021– 3033.

3. "Enhanced DOS Anomaly Detection Framework for Smart Contract Using Blockchain Metadata Analysis", International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING, SCOPUS indexed, ISSN:2147-67992147-6

Conference Papers:

1. "Cyber Attack Detection Using Bellman Optimality Equation in Reinforcement Learning", Advances in Computer Science Research- International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022), SPRINGERNATURE. DOI 10.2991/978-94-6463-136-4_19

2. "GasShield: An Advanced Detection System for Gas Consumption Attacks on Blockchain Smart Contracts", Hinweis Second International Conference on ASIT-2024. SCOPUS Indexed.

Intellectual Property:

1. Canara Copyright granted- "Detection of Dos Attack in Network Using Bellman Equation in Reinforcement Learning" 1214202, 12th June,2024

2. UK Patent granted – "Cybersecurity Threat Detection Device", Design number: 6372280, Grant date: 24 June 2024

3. Indian Patent published – "Improving Bitcoin Transaction Security: Ransomware Attack Detection using Reinforcement Learning in Blockchain", Application No. 202321040985A

Publication Date: 11/08/2023

4. Indian Copyright in the hearing stage –

“Smart Contract Blockchain Dataset”, Diary No- 15003/2023-CO/L
“A Software program for detection of DOS attack in network using Bellman’s Equation in reinforcement learning”, Diary No- 2240/2023- CO/SW